How Will Generative AI Affect Agile Software Development?

Madhu Kota[1], Kartikeya Negi[1], Balasubramaniam Ramesh[1], Lan Cao[2]
Georgia State University, USA[1] Old Dominion University, USA[2]

Research Question

How will Generative Artificial Intelligence impact the **productivity** and **human resources** (knowledge, skill, and composition) of Agile Software Development teams?

Intended audience

- Software developers and managers
- Organizational leadership

Background

- Generative AI (GenAI) has the potential to revolutionize software development more than any other recent technology [1].
- The promise of productivity improvement through GenAI has gained global attention.
- However, the impacts of GenAI in real-world software development projects and teams remain unknown.
- This research is motivated by the lack of empirical evidence regarding GenAI's effectiveness in Agile software projects.

Theoretical basis

System Dynamics:

- System dynamics provides a framework for understanding interactions and dependencies between variables in a complex system [2].
- By modeling these interactions, we simulate the potential impacts of GenAI on agile team dynamics and outcomes.

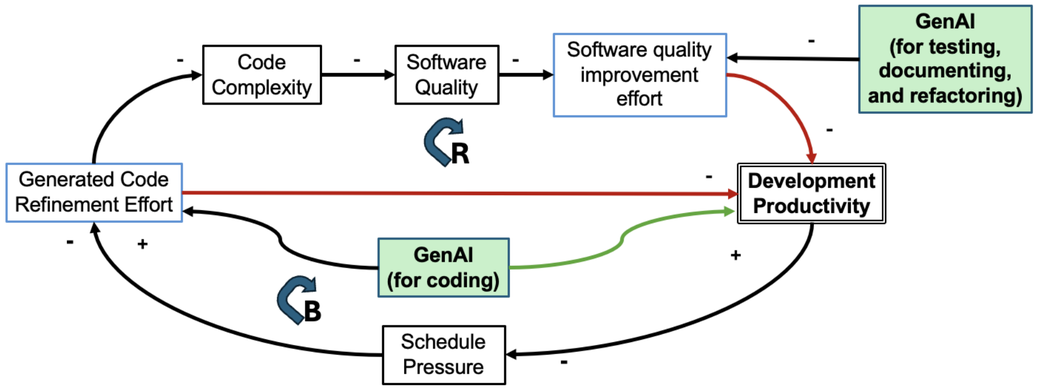
Feedback Loops:

- Positive and negative dynamic feedback loops help identify reinforcing and balancing effects of GenAI integration within the system, crucial for analyzing **short and long-term outcomes**.

Model Development

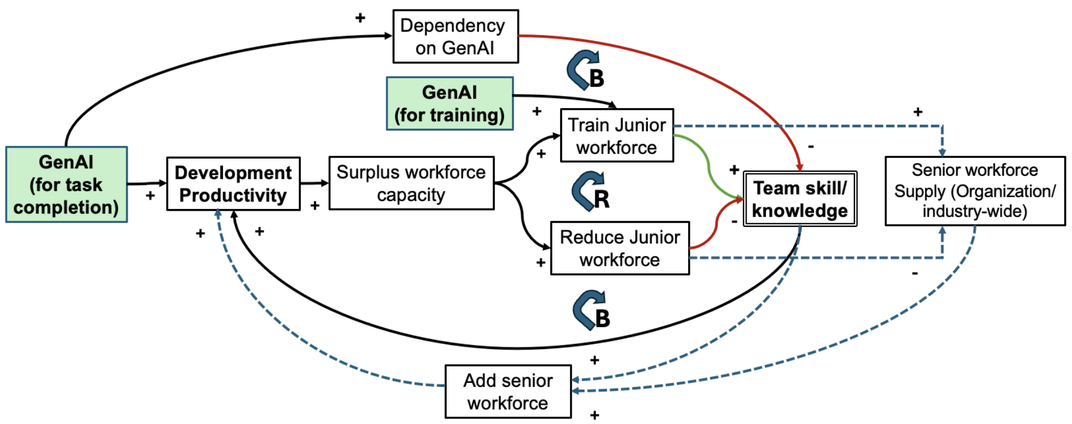
Based on prior work on agile system dynamics [3] and GenAI for software development [4], feedback loops are proposed to study the impact of GenAI in agile software development.

1. Proposed Feedback Loops of Agile Project Productivity:



- Generated code initially **increases development productivity**, but its complexity diminishes the overall quality of the software.
- **Additional human effort** is required to improve software quality, which hurts productivity.
- Using GenAI for **testing, documentation, and refactoring** can reduce human effort.

2. Proposed Feedback Loops of Human Resource management in Agile Teams:



- Dependence on GenAI can **degrade developers' skills and knowledge**.
- **Reducing junior workforce** due to productivity gains from GenAI widens the skills and knowledge gap. Instead, **Training junior developers** can reduce the gap.
- Widening of skills and knowledge gap over time leads to reduced team productivity followed by the **need to hire senior workforce**. This increases the development cost.

Data Collection and Analysis

(In Progress)

- Qualitative semi-structured interviews with software professionals who use GenAI, to modify and validate the proposed models.
- Code and analyze interview transcripts using NVivo.
- Develop final system dynamics models using Stella Architect simulation tool to simulate and forecast the impacts of GenAI on agile software development.

Expected Findings and Recommendations

1. Productivity

- Generated code initially boosts development productivity. However, it also increases code complexity, which leads to higher subsequent human effort.
- **Recommendation:** Using GenAI agents for refactoring, documentation, and testing can compensate for the additional human effort, thus increasing the overall productivity of the team.

2. Human resources

- Dependence on GenAI tools degrades developer skills and knowledge over time.
- Reduction in workforce due to increased productivity leads to further degradation of skills and knowledge, requiring hiring of senior workforce in the future.
- **Recommendation:** Retain and train junior workforce to help them gain skills, thus mitigating the need to add senior workforce in the future.

References

1. Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software, 40*(4), 30-38.
2. Abdel-Hamid, T. K. (1984). The dynamics of software development project management: An integrative system dynamics perspective *(Doctoral dissertation, Massachusetts Institute of Technology)*.
3. Cao, L., Ramesh, B., & Abdel-Hamid, T. (2010). Modeling dynamics in agile software development. *ACM Transactions on Management Information Systems (TMIS), 1*(1), 1-26.
4. Ozkaya, I. (2023). Application of large language models to software engineering tasks: Opportunities, risks, and implications. *IEEE Software, 40*(3), 4-8.

Contact information: Madhu Kota: mkota2@student.gsu.edu Kartikeya Negi: knegi1@student.gsu.edu